UNIVERSITY OF MICHIGAN ༄ SCHOOL OF INFORMATION
SI 539: Design of Complex Web Sites

Assignment 10 – Data Model and Sessions

Due Date: Monday April 14, 2008 at 11:55PM

In this assignment you will use Rails data modeling capabilities to link your chat table and members table as well as add security and login and logout capabilities to your application.

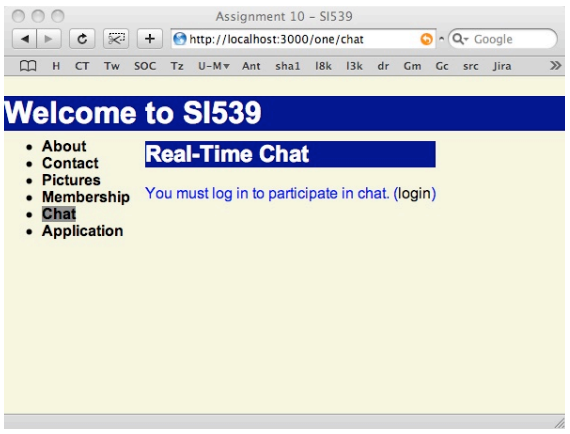
Step I: Using Session to Protect the Chat Application

Our goal will be to make sure that the chat tool only allows the posting of a chat message when you are logged in to the application.

Add the following to your chat view:

```
<% if session[:lasalle] != nil %>
<% form_remote_tag :url => 'chatcontent', :update => 'chatdiv' do -%>
   <input type="text" size="60" name="chatmsg"/>
   <%= submit_tag 'Send' %>
 <% end %>
 <% else %>
   You must log in to participate in chat.  <%= link_to "login", :action => 'login' %>
 <% end %>
</p>
```

If the user is logged in, they get the normal chat button, if not they are told to log in - navigate the chat screen - it should look as follows.


Assignment 10 – SI539
http://localhost:3000/one/chat
Google
H CT Tw SOC Tz U-M Ant sha1 I8k I3k dr Gm Gc src Jira
About
Contact
Pictures
Membership
Chat
Application

Welcome to SI539

Real-Time Chat

You must log in to participate in chat. (login)

'If there are chat messages you will see them - you just cannot send in a chat message.

Step II: Adding Login and Logout Capabilities

Add a new **login.rhtml** view as follows (this looks a lot like the join.rhtml file with some changes):

```
<h2>Please Log In</h2>
<% if flash[:notice] %><p class="notice"><%= flash[:notice] %></p><% end %>
<form method="post" action="<%= url_for :action => "login" %>">
  <fieldset>
    <legend>Required Information</legend>
   <p>
      <label for="yourmail">Enter your E-Mail:</label>
      <input type="text" name="yourmail" id="yourmail" />
    </p>
    <p>
      <label for="yourname">Enter your Password:</label>
      <input type="password" name="yourpw" id="yourpw" />
    </p>
      <input type="submit">
    </p>
  </fieldset>
</form>
<p>If you have lost your password, please send an E-Mail to
   <a href="mailto:membership@si539.com">membership@si539.com</a>
  to have your password reset.
</p>
```

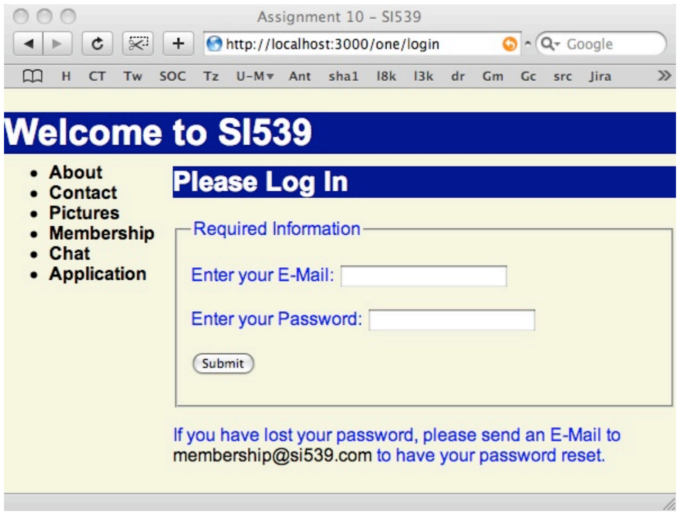
Add the following methods to your controller:

```
def login
    session[:lasalle] = nil
    if not request.post?
       return
    end

    if params[:yourpw] == nil or params[:yourmail] == nil or
     params[:yourpw] == "" or params[:yourmail] == ""
      flash[:notice] = "Please specify both E-Mail and password"
      return
    end
    memb = Member.find_by_email(params[:yourmail])
    logger.info "Retrieved member $#(memb}"
    if memb == nil or params[:yourpw] != 'secret'
      flash[:notice] = "Account / Password combination not found"
      return
    end
    session[:lasalle] = memb
    logger.info "User logged in:#{memb.email}"
    if session[:lastaction] != nil
      redirect_to :action => session[:lastaction]
      session[:lastaction] = nil
    else
      redirect_to :action => 'index'
    end
  end

  def logout
    session[:lasalle] = nil
    if session[:lastaction] != nil
      redirect_to :action => session[:lastaction]
      session[:lastaction] = nil
    else
      redirect_to :action => 'index'
    end
  end
```

If you navigate to **http://localhost:3000/One/login** or press the login button in chat you should see the following screen:


Assignment 10 – SI539
http://localhost:3000/one/login
Google
H CT Tw SOC Tz U-M Ant sha1 I8k I3k dr Gm Gc src Jira
About
Contact
Pictures
Membership
Chat
Application
Submit

Welcome to SI539

Please Log In

Required Information

Enter your E-Mail:

Enter your Password:

If you have lost your password, please send an E-Mail to membership@si539.com to have your password reset.

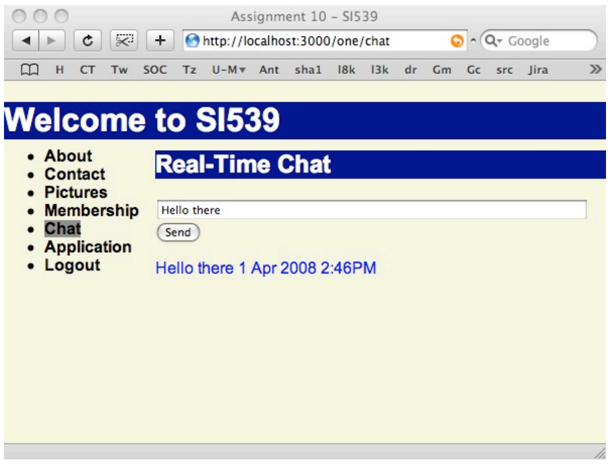
The password is always "secret" - the E-Mail address must exist in the member table. If you want you can add a password field to your table and look up both the ID and PW in the table.

Make sure that the flash message comes up when the account or password is wrong. Once you enter a proper mail address and password - you should come back to Chat with a screen to send Chat messages.

To display a logout button, make the following changes to your **_nav.rhtml** file:

```
<div id="navigation">
<ul>
<%= do_nav_entry("About","index") %>
<%= do_nav_entry("Contact","contact") %>
<%= do_nav_entry("Pictures","pictures") %>
<%= do_nav_entry("Membership","members") %>
<%= do_nav_entry("Chat","chat") %>
<%= do_nav_entry("Application","join") %>
<% if session[:lasalle] != nil %>
<%= do_nav_entry("Logout","logout") %>
<% end %>
</ul>
<%
  if params[:action] != 'login' and params[:action] != 'logout'
     session[:lastaction] = params[:action]
  end
%>
</div>
```

Now once you are logged in the "Logout" option should appear in the navigation as follows:


Assignment 10 – SI539
http://localhost:3000/one/chat
Google
H CT Tw SOC Tz U-M▾ Ant sha1 I8k I3k dr Gm Gc src Jira
Send

Welcome to SI539

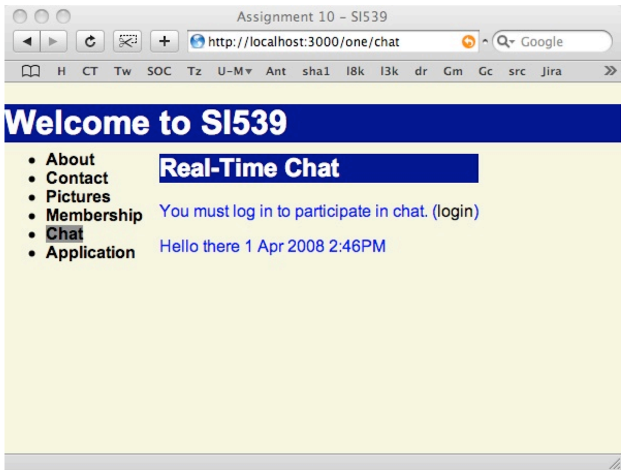
- About
- Contact
- Pictures
- Membership
- Chat
- Application
- Logout

Real-Time Chat

Hello there

Hello there 1 Apr 2008 2:46PM

If you press the **Logout** navigation button, the logout should disappear and then you should come back to whichever page you were on before you pressed "Logout".


Assignment 10 – SI539
http://localhost:3000/one/chat
Google
H CT Tw SOC Tz U-M▾ Ant sha1 I8k I3k dr Gm Gc src Jira

Welcome to SI539

- About
- Contact
- Pictures
- Membership
- Chat
- Application

Real-Time Chat

You must log in to participate in chat. (login)

Hello there 1 Apr 2008 2:46PM

Logout form several different screens to verify that logout actually goes back to whatever screen you were on when you pressed logout.

Part III: Add A Data Model Relationship

Delete all of the records from the chat table suing the SQLite Browser. Make sure you are in the assignment 10 directory ☺. Save the database file. Go into chat to verify that there are no chat messages displayed.

Go into the chat.rb data model file and make the following change:

```
class Chat < ActiveRecord::Base
  belongs_to :member
end
```

The name "member" must match the name of the other model (in this case the other data model file is named **member.rb**). Don't include the '.rb' - just the name of the model - with the first character lower case.

Yout chat table should have a column named member_id which is an integer (this was part of the previous assignment).

```
class CreateChats < ActiveRecord::Migration
  def self.up
    create_table :chats do |t|
      t.column :chatmsg, :string
      t.column :member_id, :integer
      t.column :created_at, :datetime
    end
  end

  def self.down
    drop_table :chats
  end
end
```

The name of the column must match the name of the model (member) with "_id" tacked on to the end. And the column must be an integer.

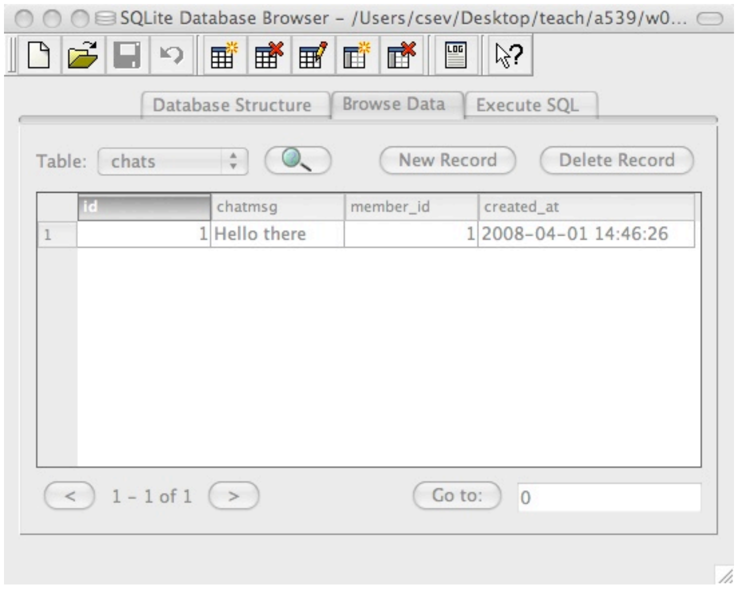
Save the model file and stop and restart your **ruby script/server** window. Make sure that you have no errors on startup due to this change - these model files are read when the server first comes up.

Now go into your controller and add the following to the **chatcontent** method in your controller.

```
def chatcontent
   if request.post? and params[:chatmsg] != nil and session[:lasalle] != nil
      logger.info "Chat"
     ch = Chat.new
     ch.chatmsg = params[:chatmsg]
     ch.member = session[:lasalle]
     ch.save
   end
   @chats = Chat.find(:all, :order => "chats.created_at DESC",
        :limit => 5)
   logger.info "We found #{@chats.size} chats"
   render :action => 'chatcontent', :layout => false
end
```

This records the current logged in member in the chat record each time a new record is created. It also demands that someone is logged in (`session[:lasalle] != nil`) before adding the chat message.

Go ahead and add a chat message - and then take a look at the database in SQLIte Browser nd verify that the Member Id is properly set in the new chat record:

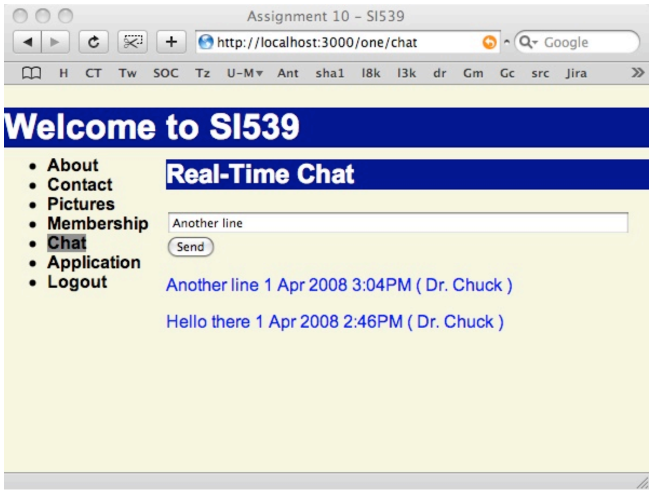


	id	chatmsg	member_id	created_at
1	1	Hello there	1	2008-04-01 14:46:26

Make the following changes to the **chatcontent** view:

```
<% for chat in @chats %>
<p class="chattext"><%= chat.chatmsg %>
<span class="chatdate">
<%= chat.created_at.strftime("%e %b %Y %l:%M%p") %>
<% if chat.member != nil %>
  ( <%= chat.member.name %> )
<% end %>
</span>
</p>
<% end %>
```

This adds the member name to the chat output and your chat screen should look as follows:



Another line 1 Apr 2008 3:04PM (Dr. Chuck)

Hello there 1 Apr 2008 2:46PM (Dr. Chuck)

You should see the member name at the end of each chat line.

Hand In

Hand in screen shots equivalent to all of the above screen shots for your application.

Also hand in your controller.rb, _nav.rhtml, login.rhtml, chat.rhtml, and chatcontent.rhtml.